

I. Amendments to the Claims

Please amend the claims as follows with the following version of the claims in accordance with revised 37 CFR § 1.121.

1. (Currently Amended) A method for processing data within a distributed data processing system, the method comprising:

receiving, at a client, a first file in response to a request ~~to~~ by a user to browse the first file;

5 displaying content from the first file by a client application;

selecting a user interface control within the client application;

10 in response to the selection of the user interface control, automatically retrieving an address of a server, wherein the user has previously established a user account at the server;

15 in response to the selection of the user interface control, automatically retrieving user-specified parameters within the client application, wherein the user-specified parameters are associated with the user account at the server for server-side processing of files sent by the user to the server; and

automatically sending the first file with the retrieved user-specified parameters from the client to the server using the retrieved address.

2. (Original) The method of claim 1 further comprising:
determining that the first file references a plurality of files;

receiving at the client the plurality of files;

25 sending the plurality of files with the first file to the server.

3. (Original) The method of claim 1 wherein the address is a Uniform Resource Identifier (URI).

30 4. (Original) The method of claim 1 wherein the first file is formatted in accordance with a markup language.

5. (Original) A method for processing data within a distributed data processing system, the method comprising:
receiving, at a server, one or more files from a user at a client, wherein the user has previously established a user
5 account at the server;
authorizing the user for processing data at the server;
in response to authorizing the user, automatically storing the one or more received files from the client at the server;
in response to authorizing the user, automatically
10 retrieving a Web page from local storage at the server;
automatically modifying the retrieved Web page by inserting a hyperlink to at least one of the one or more received files from the client; and
automatically storing the modified Web page.

15 6. (Original) The method of claim 5 wherein the Web page is retrieved from local storage at the server.

20 7. (Original) The method of claim 5 wherein the Web page may be edited by the user.

8. (Original) The method of claim 5 further comprising:
parsing at least one received file from the client to retrieve an originating Uniform Resource Identifier (URI);
25 generating one or more URIs for storing the one or more received files at the server; and
storing the one or more received files to be accessible using the one or more generated URIs.

30 9. (Original) The method of claim 8 wherein the inserted hyperlink references a received file using a generated URI.

10. (Original) The method of claim 8 wherein the inserted hyperlink is associated with anchor text derived from content within a received file.

5 11. (Original) The method of claim 10 wherein the anchor text is a title of a received file.

12. (Original) The method of claim 5 further comprising:
executing a server-side script against the one or more
10 received files and the retrieved Web page.

13. (Original) The method of claim 12 further comprising:
determining whether a user has specified a server-side
script; and

15 in response to a determination that the user has specified a server-side script, executing the specified server-side script.

14. (Original) The method of claim 13 further comprising:
parsing at least one received file from the client to
20 retrieve the specified server-side script.

15. (Original) The method of claim 5 further comprising:
parsing at least one received file from the client to
retrieve a user-specified processing parameter, wherein the
25 user-specified processing parameter identifies the Web page to be retrieved.

16. (Currently Amended) An apparatus for processing data within a distributed data processing system, the apparatus comprising:

first receiving means for receiving, at a client, a first file in response to a request ~~to~~ by a user to browse the first file;

displaying means for displaying content from the first file by a client application;

selecting means for selecting a user interface control within the client application;

first retrieving means for automatically retrieving, in response to the selection of the user interface control, an address of a server, wherein the user has previously established a user account at the server;

second retrieving means for automatically retrieving, in response to the selection of the user interface control, user-specified parameters within the client application, wherein the user-specified parameters are associated with the user account at the server for server-side processing of files sent by the user to the server; and

first sending means for automatically sending the first file with the retrieved user-specified parameters from the client to the server using the retrieved address.

17. (Original) The apparatus of claim 16 further comprising:
determining means for determining means for determining that the first file references a plurality of files;

second receiving means for receiving at the client the plurality of files;

second sending means for sending the plurality of files with the first file to the server.

18. (Original) The apparatus of claim 16 wherein the address is a Uniform Resource Identifier (URI).

5 19. (Original) The apparatus of claim 16 wherein the first file is formatted in accordance with a markup language.

20. (Original) An apparatus for processing data within a distributed data processing system, the apparatus comprising:
receiving means for receiving, at a server, one or more files from a user at a client, wherein the user has previously
5 established a user account at the server;
authorizing means for authorizing the user for processing data at the server;
first storing means for automatically storing, in response to authorizing the user, the one or more received files from the
10 client at the server;
retrieving means for automatically retrieving, in response to authorizing the user, a Web page from local storage at the server;
modifying means for automatically modifying the retrieved
15 Web page by inserting a hyperlink to at least one of the one or more received files from the client; and
second storing means for automatically storing the modified Web page.

20 21. (Original) The apparatus of claim 20 wherein the Web page is retrieved from local storage at the server.

22. (Original) The apparatus of claim 20 wherein the Web page may be edited by the user.

25 23. (Original) The apparatus of claim 20 further comprising:
first parsing means for parsing at least one received file from the client to retrieve an originating Uniform Resource Identifier (URI);

30 generating means for generating one or more URIs for storing the one or more received files at the server; and

third storing means for storing the one or more received files to be accessible using the one or more generated URIs.

24. (Original) The apparatus of claim 23 wherein the inserted hyperlink references a received file using a generated URI.

5

25. (Original) The apparatus of claim 23 wherein the inserted hyperlink is associated with anchor text derived from content within a received file.

10 26. (Original) The apparatus of claim 25 wherein the anchor text is a title of a received file.

27. (Original) The apparatus of claim 20 further comprising:
first executing means for executing a server-side script
15 against the one or more received files and the retrieved Web page.

28. (Original) The apparatus of claim 27 further comprising:
determining means for determining whether a user has
20 specified a server-side script; and
second executing means for executing in response to a determination that the user has specified a server-side script, the specified server-side script.

25 29. (Original) The apparatus of claim 28 further comprising:
second parsing means for parsing at least one received file from the client to retrieve the specified server-side script.

30 30. (Original) The apparatus of claim 20 further comprising:
third parsing means for parsing at least one received file from the client to retrieve a user-specified processing parameter, wherein the user-specified processing parameter identifies the Web page to be retrieved.

31. (Currently Amended) A computer program product in a computer readable medium for use in a data processing system for remotely storing data, the computer program product comprising:

instructions for receiving, at a client, a first file in response to a request ~~to~~ by a user to browse the first file;
instructions for displaying content from the first file by a client application;

instructions for selecting a user interface control within the client application;

instructions for automatically retrieving, in response to the selection of the user interface control, an address of a server, wherein the user has previously established a user account at the server;

instructions for automatically retrieving, in response to the selection of the user interface control, user-specified parameters within the client application, wherein the user-specified parameters are associated with the user account at the server for server-side processing of files sent by the user to the server; and

instructions for automatically sending the first file with the retrieved user-specified parameters from the client to the server using the retrieved address.

32. (Original) The computer program product of claim 31 further comprising:

instructions for determining that the first file references a plurality of files;

instructions for receiving at the client the plurality of files;

instructions for sending the plurality of files with the first file to the server.

33. (Original) The computer program product of claim 31 wherein the address is a Uniform Resource Identifier (URI).

5 34. (Original) The computer program product of claim 31 wherein the first file is formatted in accordance with a markup language.

35. (Original) A computer program product in a computer readable medium for use in a data processing system for storing data, the computer program product comprising:

instructions for receiving, at a server, one or more files from a user at a client, wherein the user has previously established a user account at the server;

instructions for authorizing the user for processing data at the server;

instructions for automatically storing, in response to authorizing the user, the one or more received files from the client at the server;

instructions for automatically retrieving, in response to authorizing the user, a Web page from local storage at the server;

instructions for automatically modifying the retrieved Web page by inserting a hyperlink to at least one of the one or more received files from the client; and

instructions for automatically storing the modified Web page.

36. (Original) The computer program product of claim 35 wherein the Web page is retrieved from local storage at the server.

37. (Original) The computer program product of claim 35 wherein the Web page may be edited by the user.

38. (Original) The computer program product of claim 35 further comprising:

instructions for parsing at least one received file from the client to retrieve an originating Uniform Resource Identifier (URI);

instructions for generating one or more URIs for storing the one or more received files at the server; and

instructions for storing the one or more received files to be accessible using the one or more generated URIs.

39. (Original) The computer program product of claim 38 wherein the inserted hyperlink references a received file using a generated URI.

40. (Original) The computer program product of claim 38 wherein the inserted hyperlink is associated with anchor text derived from content within a received file.

41. (Original) The computer program product of claim 40 wherein the anchor text is a title of a received file.

42. (Original) The computer program product of claim 35 further comprising:

instructions for executing a server-side script against the one or more received files and the retrieved Web page.

43. (Original) The computer program product of claim 42 further comprising:

instructions for determining whether a user has specified a server-side script; and

instructions for executing, in response to a determination that the user has specified a server-side script, the specified server-side script.

44. (Original) The computer program product of claim 43
further comprising:

5 instructions for parsing at least one received file from the
client to retrieve the specified server-side script.

45. (Original) The computer program product of claim 35
further comprising:

10 instructions for parsing at least one received file from the
client to retrieve a user-specified processing parameter, wherein
the user-specified processing parameter identifies the Web page
to be retrieved.

II. General Remarks Concerning This Response

Claims 1-45 are currently pending in the present application. Claims 1, 16, and 31 have been amended; no claims have been added or canceled in this response. Reconsideration of the claims is respectfully requested.

Independent claims 1, 16, and 31 have been amended to correct typographical errors in which the word "to" inadvertently appeared and the word "with" inadvertently did not appear. Applicant notes that these amendments are minor in nature and have not been made in consideration of prior art.

III. Summary of Present Invention

A method, system, apparatus, and computer program product are presented for customizing the storage of captured Web content. The client receives a file, generally a Web page, in response to a request to by a user to browse the Web page. The Web page may be displayed by a browser application, and the user may select a user interface control within the browser that indicates that the user desires to capture the content being displayed by the browser and push the content to the server for customized processing. The browser automatically retrieves an address of a server at which the user has previously established a user account for the capture service. Other user-specified parameters can also be retrieved by the browser for sending to the server. The captured data and user parameters are sent to the server.

The server receives the data, and assuming that the user is authorized for processing data at the server, the server automatically stores the captured data received from the client at the server. In addition, the server automatically retrieves a Web page and automatically modifies the retrieved Web page by inserting a hyperlink to the captured data received from the client. The server then automatically stores the modified Web

page. Preferably, the server executes server-side scripts for modifying the Web page. In this manner, the user can customize the manner in which the server processes the Web page and/or the captured data so that the hyperlinks are available in a manner preferred by the user.

IV. 35 U.S.C. § 102(e)-Anticipation-Khan

The Office action has rejected claims 1-45 under 35 U.S.C. § 102(e) as anticipated by Khan, "System method and article of manufacture for dynamically user-generated Internet search directory based on prioritized server-sided user bookmarks", U.S. Patent No. 6,546,393 B1, filed 10/07/1999, issued 04/08/2003. This rejection is respectfully traversed.

Independent claim 1, as amended, reads as follows:

1. A method for processing data within a distributed data processing system, the method comprising:
receiving, at a client, a first file in response to a request by a user to browse the first file;
displaying content from the first file by a client application;
selecting a user interface control within the client application;
in response to the selection of the user interface control, automatically retrieving an address of a server, wherein the user has previously established a user account at the server;
in response to the selection of the user interface control, automatically retrieving user-specified parameters within the client application, wherein the user-specified parameters are associated with the user account at the server for server-side processing of files sent by the user to the server; and
automatically sending the first file with the retrieved user-specified parameters from the client to the server using the retrieved address.

Applicant notes that the first element in claim 1 recites an operation to retrieve a file for browsing, e.g., such as retrieving a web page that is identified by a Uniform Resource Locator (URL), and the second element of claim 1 recites the display of the file, e.g., such as a browser application

presenting a retrieved web page on a display device. The third element of claim 1 recites that a user selects a control, and the fourth element of claim 1 recites that an address of a server is obtained in response to the user action; in addition, the user
5 has a user account at the server for performing authorized remote operations at the server. The rejection of claim 1 addresses each of these elements of claim 1 with an anticipation argument that references various portions of Khan, and Applicant does not object to the analysis of Khan with respect to the first through
10 fourth elements of claim 1.

However, the fifth element of claim 1 reads:

in response to the selection of the user interface control, automatically retrieving user-specified parameters within the client application, wherein the user-specified
15 parameters are associated with the user account at the server for server-side processing of files sent by the user to the server;

The rejection states that Khan discloses this feature in Figure
20 6, which illustrates an online bookmark account page, and column 14, lines 32-43 of Khan, which reads:

In an embodiment of the present invention, as soon as a user signs up, the user may be set up with a personal, password-protected, web-based online bookmark account.

Every time the user accesses the Internet, wherever the
25 user is, the user is able to sign in and see the user's complete bookmarks selection. No matter what browser, no matter which operating system, the user's favorite links are there via the user's online bookmark account for the user to
30 use.

FIG. 6 illustrates an exemplary display of an online bookmark account page 600 of a user. This exemplary page 600 has a central area 602, a left menu bar 604 and a top menu bar 606. Also displayed may be a bookmark adding area
35 608.

Apparently, the rejection argues that "the user's complete bookmarks selection", i.e. "the user's favorite links", in Khan represents "the user-specified parameters" as recited in the
40 claim. However, the rejection does not specifically address each feature within the fourth element of claim 1, specifically the

phrase "for server-side processing of files sent by the user to the server". At this point, Applicant merely notes this deficiency in the rejection because a similar deficiency exists in the manner in which the rejection has addressed the sixth element of claim 1.

The sixth element of claim 1 reads:

automatically sending the first file with the retrieved user-specified parameters from the client to the server using the retrieved address.

The rejection states this feature is disclosed in Figure 27 of Khan, which illustrates a pop-up window that lets a user add and file bookmarks to their user account while they are surfing the Internet without having to return to their online bookmarks account page, and in Figure 11 of Khan, which illustrates a browser application window that is displaying an online bookmarks account page at a web site with a URL of "http://www.clickmarks.com". More specifically, the rejection states that the claimed feature of "automatically sending the first file" is disclosed when the user employs the pop-up window in Figure 27 to send the URL of "http://www.delphi.com" to the server at "http://www.clickmarks.com" while the browser application window in the background displays a web page at address "http://www.delphi.com" as shown in the "Address" location field of the browser application window.

However, Figure 27 in Khan clearly shows an "Add URL" button 2704 "to execute this function and add a bookmark for this site to the user's account."--(Khan, column 22, lines 59-61). Hence, the portion of Khan that has been applied by the rejection against the sixth element of claim 1 clearly discloses that, when requested by a user through a user action, only a URL is sent to the server that maintains the online bookmark accounts, e.g., the text string "http://www.delphi.com", not the web page that is being viewed when the corresponding URL is sent, e.g., the web page that corresponds to the URL "http://www.delphi.com".

Applicant asserts that the rejection relies either on a severe misinterpretation of the claim language or on a severe misinterpretation of Khan. Khan clearly discloses that only URLs or bookmarks are sent from a client to a server. In contrast, claim 1 of the present patent application clearly recites in its sixth element that the file, i.e. the file that was recited in the first element as being previously retrieved for browsing, is sent from the client to the server. In addition, as noted above, the fifth element of claim 1 clearly recites that the user-specified parameters in the user account at the server specifically include parameters for server-side processing of files sent by the user to the server. In other words, the present invention has features that include the transmission, to a server, of one or more files that are being browsed by a user, whereas Khan has features that include the transmission, to a server, of one or more URLs that the user may or may not be concurrently browsing. Khan does not disclose "sending the first file ... to the server" as required by the claim language of claim 1.

Moreover, the argument in the rejection of claim 1 is internally inconsistent. The rejection argues that "the first file" in the first element of claim 1 refers to the web pages that are shown in Figures 9, 13, 14, or 27. However, the rejection argues that "the first file" in the sixth element of claim 1 only refers to the URL, e.g., "www.delphi.com", of a web page. One having ordinary skill in the art would not consider a web page to be a browsed file and also consider the URL of the web page to be a browsed file; it is illogical for the rejection to argue that the URLs that are sent to the server, as in Khan, are equivalent to the web pages that are being browsed in Khan.

Additional support for Applicant's argument, i.e. that only URLs are sent from the client to the server in the system that is disclosed in Khan, can be found elsewhere within Khan. Khan

states in column 15, line 60, to column 16, line 14 (emphasis added):

Remote Addition

5 With particular reference to FIG. 9, a user may also
add **bookmarks** to their online bookmark account without even
being on the online bookmark provider's web page by using an
"Adding Bookmarks Remotely" feature 902 that may be included
on a user's web browser 900. **This feature lets a user add**
10 **links while surfing on the Internet by clicking on the**
function 902 as shown in FIG. 9.

Accessing Bookmarks Added through the "Adding Bookmarks Remotely" Feature

15 All the URLs that users add to their current online
bookmark account through the "Adding Bookmarks Remotely"
feature may be stored in their online bookmark account and
are accessible from any computer on the Internet, just like
all their other bookmarks in their online bookmark account.
When a user wishes to bookmark a particular page using this
function, the user first accesses the particular page
20 through their browser. Once the user is there at the
particular page, the user may add a bookmark remotely by
clicking on, for example, the ADD2My_Stuff favorite/bookmark
in their current browser. **The URL of the particular page is**
then stored in their My_Stuff folder.

25 Khan also states at column 16, line 43, to column 17, line 16
(emphasis added):

Adding Directly

30 With reference to FIG. 12, a user may add **bookmarks by**
typing the URL (web site address) into the box 1200 next to
"New Link" and then clicking on "Add" 702. A user may add
folders in the same way, by typing the new folder's name
into the box 1204 next to "New Folder" and then clicking on
"Add" 702. Both new bookmarks and new folders may be placed
35 inside other folders. A user may do this by clicking on the
down arrow 1206 on the "choose location of new url/folder"
box 1208 and selecting the folder 1210 users want to move
the new item into.

Adding Links Remotely

40 A user does not have to be in their account to add
bookmarks. Users may actually add them while they surf the
Internet. First, the user needs to set up the Adding Links
Remotely function. With reference to FIG. 13, once the
Adding Bookmarks Remotely function is set up, the user may
45 see a new bookmark called "Add2Clickmarks" 1300 on their
browser 1302. This means that the Netscape Bookmarks list
or their Internet Explorer Favorites list will have this new

"Add2Clickmarks" bookmark. While a user surfs the Internet, if the user finds a web site he or she wants to add to their account, the user just clicks on the browser bookmark called "Add2Clickmarks" and the site will be automatically added to their online bookmark account.

Adding Links through a Pop-Up Window

To add links to a user account without coming back to the user account page, a user may use the Floating Link Adder pop-up window 1400 which is illustrated in FIG. 14. This feature is useful when using the present invention from a computer that is not the user's regular workstation.

With reference to FIG. 7, a Floating Link Adder button 722 may be clicked on from a user account page for displaying of the Floating Link Adder pop-up window 1400 illustrated in FIG. 14. In the pop-up window 1400, the user may then input in box 1402 the URL that they want to add. By default, the new link will be added to the top level folder in the user's account. The user may also choose the specific folder that users want the new link to be added to by selection of a folder displayed in box 1404. Clicking on the Add URL button 1406 adds the link to the user's account.

The rejection clearly misinterprets the features of Khan and/or improperly applies those features to the claim elements of the present invention.

In addition, Applicant distinguished the present invention from online bookmark services such as the one that is disclosed in Khan. On page 10 of the specification of the present patent application, Applicant states:

Various portal services are operated on the World Wide Web and generally contain large amounts of information. In order to attract users to view advertisements on Web pages, some portals offer the ability of users to store information on a server so that the information is available to the user from more than one client device, which is particularly useful if the user accesses the Internet from more than one machine.

A user at client 202 may store personal bookmark file 242 within storage 244 connected to portal Web server 246. By storing or replicating a copy of a browser bookmark file on the server, the user can have access to the bookmarks from almost any Web-enabled device throughout the Internet.

Such online bookmark services were well-known at the time of the present invention, and Applicant has taken care to describe the

differences between these services and the present invention. On page 12 of the specification of the present patent application, Applicant states an advantage of the present invention:

5 The present invention provides a methodology by which a user may browse Web pages and then, when desired, capture those Web pages for later viewing, editing, or other processing in a customized manner. By selecting an application control within browser 304, a user can request that browser 304 push Web page 306 to server 332, which
10 stores the captured Web page in server storage 334. It should be noted that other types of content may also be captured other than Web pages, such as graphic files, text files, audio and video files, general binary data files, etc.

15 The errors and logical inconsistencies in the rejection of independent claim 1 are continued in the rejection of independent claim 5, which includes similar claim elements as in claim 1, except that claim 5 is directed to a process from the perspective
20 of a server whereas claim 1 is directed to a process from the perspective of a client. For example, the first element of independent claim 5 recites that a server receives one or more files from a client, not simply URLs from a client as disclosed in Khan.

25 Independent claims 1 and 5 and their dependent claims are directed to a method; independent claims 16 and 20 and their dependent claims are directed to an apparatus; and independent claims 31 and 35 and their dependent claims are directed to a computer program product. The Office action presents a similar
30 anticipation argument against claims 16, 20, 31, and 35 as was used against claims 1 and 5. Applicant's argument with respect to the rejection of claims 1 and 5 is similarly applicable against the rejection of claims 16, 20, 31, and 35.

35 Khan clearly does not disclose features as required by the language of the claims of the present application. As stated at MPEP § 2131: "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or